```
 / _____)             _              | |
( (____  _____ ____ _| |_ _____  ____| |__
 \____ \| ___ |    (_   _) ___ |/ ___)  _ \
 _____) ) ____| | | || |_| ____( (___| | | |
(______/|_____)_|_|_| \__)_____)\____)_| |_|
  (C)2013 Semtech-Cycleo
```

Basic communication protocol between Lora gateway and server

1. Introduction

The protocol between the gateway and the server is purposefully very basic and for demonstration purpose only, or for use on private and reliable networks.

There is no authentication of the gateway or the server, and the acknowledges are only used for network quality assessment, not to correct UDP datagrams losses (no retries).

2. System schematic and definitions

```
((( Y )))
    |
    |
+- -|- - - - - - - - - - - - -+        xxxxxxxxxxxx          +--------+
|+--+-----------+     +------+|       xx x  x     xxx        |        |
||              |     |      ||      xx  Internet  xx        |        |
|| Concentrator |<----+ Host |<------xx     or     xx-------->|        |
||              | SPI |      ||      xx  Intranet  xx        | Server |
|+--------------+     +------+|       xxxx   x   xxxx        |        |
|                             |           xxxxxxxx           |        |
|            Gateway          |                              |        |
+- - - - - - - - - - - - - - -+                              +--------+
```

Concentrator: radio RX/TX board, based on Semtech multichannel modems (SX130x), transceivers (SX135x) and/or low-power stand-alone modems (SX127x).

Host: embedded computer on which the packet forwarder is run. Drives the concentrator through a SPI link.

Gateway: a device composed of at least one radio concentrator, a host, some network connection to the internet or a private network (Ethernet, 3G, Wifi, microwave link), and optionally a GPS receiver for synchronization.

Server: an abstract computer that will process the RF packets received and forwarded by the gateway, and issue RF packets in response that the gateway will have to emit.

It is assumed that the gateway can be behind a NAT or a firewall stopping any incoming connection.
It is assumed that the server has an static IP address (or an address solvable through a DNS service) and is able to receive incoming connections on a specific port.

3. Upstream protocol

3.1. Sequence diagram

```
+---------+                                                    +---------+
| Gateway |                                                    | Server  |
+---------+                                                    +---------+
     | -----------------------------------\                         |
     |-| When 1-N RF packets are received |                         |
     | ------------------------------------                         |
     |                                                              |
     | PUSH_DATA (token X, GW MAC, JSON payload)                    |
     |------------------------------------------------------------->|
     |                                                              |
     |                                           PUSH_ACK (token X) |
     |<-------------------------------------------------------------|
     |                              ------------------------------\ |
     |                              | process packets *after* ack |-|
     |                              ------------------------------- |
     |                                                              |
```

3.2. PUSH_DATA packet

That packet type is used by the gateway mainly to forward the RF packets received, and associated metadata, to the server.

Bytes	Function
0	protocol version = 1
1-2	random token
3	PUSH_DATA identifier 0x00
4-11	Gateway unique identifier (MAC address)
12-end	JSON object, starting with {, ending with }, see section 4

3.3. PUSH_ACK packet

That packet type is used by the server to acknowledge immediately all the PUSH_DATA packets received.

Bytes	Function
0	protocol version = 1
1-2	same token as the PUSH_DATA packet to acknowledge
3	PUSH_ACK identifier 0x01

4. Upstream JSON data structure

The root object must contain an array named "rxpk":

```
{
    "rxpk":[ {...}, ...]
}
```

That array contains at least one JSON object, each object contain a RF packet and associated metadata with the following fields:

Name	Type	Function
time	string	UTC time of packet reception, us precision, ISO 8601 format
tmst	number	Internal timestamp of "RX finished" event (32b unsigned)
freq	number	RX central frequency in MHz (unsigned float, Hz precision)
chan	number	Concentrator "IF" channel used for RX (unsigned integer)
rfch	number	Concentrator "RF chain" used for RX (unsigned integer)
stat	number	CRC status: 1 = OK, -1 = fail, 0 = no CRC
modu	string	Modulation identifier "LORA" or "FSK"
datr	string	Datarate identifier (eg. SF12BW500 for Lora)
codr	string	ECC coding rate identifier
rssi	number	RSSI in dBm (signed integer, 1 dB precision)
lsnr	number	Lora SNR ratio in dB (signed float, 0.1 dB precision)
size	number	RF packet payload size in bytes (unsigned integer)
data	string	Base64 encoded RF packet payload, no padding

Example (white-spaces, indentation and newlines added for readability):

```
{
    "rxpk":[
        {
            "time":"2013-03-31T16:21:17.528002Z",
            "tmst":3512348611,
            "chan":2,
            "rfch":0,
            "freq":866.349812,
            "stat":1,
            "modu":"LORA",
            "datr":"SF7BW125",
            "codr":"4/6",
            "rssi":-35,
            "lsnr":5.1,
            "size":32,
            "data":"-DS4CGaDCdG+48eJNM3Vai-zDpsR71Pn9CPA9uCON84"
        },{
            "time":"2013-03-31T16:21:17.532038Z",
            "tmst":3316387610,
            "chan":0,
            "rfch":0,
            "freq":863.00981,
            "stat":1,
            "modu":"LORA",
            "datr":"SF10BW125",
            "codr":"4/7",
            "rssi":-38,
            "lsnr":5.5,
            "size":32,
            "data":"ysgRl452xNLep9S1NTIg2lomKDxUgn3DJ7DE+b00Ass"
        }
    ]
}
```

5. Downstream protocol

5.1. Sequence diagram

```
+---------+                                                  +---------+
| Gateway |                                                  | Server  |
+---------+                                                  +---------+
     | -----------------------------------\                       |
     |-| Every N seconds (keepalive time) |                       |
     | ------------------------------------                         |
     |                                                              |
     | PULL_DATA (token Y, MAC@)                                    |
     |------------------------------------------------------------->|
     |                                                              |
     |                                           PULL_ACK (token Y) |
     |<-------------------------------------------------------------|
     |                                                              |

+---------+                                                    +---------+
| Gateway |                                                    | Server  |
+---------+                                                    +---------+
     |      ------------------------------------------------------\ |
     |      | Anytime after first PULL_DATA for each packet to TX |-|
     |      -------------------------------------------------------  |
     |                                                              |
     |                            PULL_RESP (token 0, JSON payload) |
     |<-------------------------------------------------------------|
     |                                                              |
```

5.2. PULL_DATA packet

That packet type is used by the gateway to poll data from the server.

This data exchange is initialized by the gateway because it might be impossible for the server to send packets to the gateway if the gateway is behind a NAT.

When the gateway initialize the exchange, the network route towards the server will open and will allow for packets to flow both directions.
The gateway must periodically send PULL_DATA packets to be sure the network route stays open for the server to be used at any time.

Bytes	Function
0	protocol version = 1
1-2	random token
3	PULL_DATA identifier 0x02
4-11	Gateway unique identifier (MAC address)

5.3. PULL_ACK packet

That packet type is used by the server to confirm that the network route is open and that the server can send PULL_RESP packets at any time.

Bytes	Function
0	protocol version = 1
1-2	same token as the PULL_DATA packet to acknowledge
3	PULL_ACK identifier 0x04

5.4. PULL_RESP packet

That packet type is used by the server to send RF packets and associated metadata that will have to be emitted by the gateway.

Bytes	Function
0	protocol version = 1
1-2	unused bytes
3	PULL_RESP identifier 0x03
4-end	JSON object, starting with {, ending with }, see section 6

6. Downstream JSON data structure

The root object must contain an object named "txpk":

```
{
    "txpk": {...}
}
```

That object contain a RF packet to be emitted and associated metadata with the following fields:

Name	Type	Function
imme	bool	Send packet immediately (will ignore tmst & time)
tmst	number	Send packet on a certain timestamp value (will ignore time)

time	string	Send packet at a certain time (GPS synchronization required)
freq	number	TX central frequency in MHz (unsigned float, Hz precision)
rfch	number	Concentrator “RF chain” used for RX (unsigned integer)
powe	number	TX output power in dBm (unsigned integer, dBm precision)
modu	string	Modulation identifier “LORA” or “FSK”
datr	string	Datarate identifier (eg. SF12BW500 for Lora)
codr	string	ECC coding rate identifier
ipol	bool	Lora modulation polarization inversion
prea	number	RF preamble size (unsigned integer)
size	number	RF packet payload size in bytes (unsigned integer)
data	string	Base64 encoded RF packet payload, no padding

Most fields are optional.
If a field is omitted, default parameters will be used.

Example (white-spaces, indentation and newlines added for readability):

```
{
    "txpk":{
        "imme":true,
        "freq":864.123456,
        "rfch":0,
        "powe":14,
        "modu":"LORA",
        "datr":"SF11BW125",
        "codr":"4/6",
        "ipol":false,
        "size":32,
        "data":"H3P3N2i9qc4yt7rK7ldqoeCVJGBybzPY5h1Dd7P7p8v"
    }
}
```